

DNA-SeAl: Sensitivity Levels to Optimize the Performance of Privacy-Preserving DNA Alignment

Maria Fernandes, Jérémie Decouchant, Marcus Völz, Member, IEEE,
Francisco M Couto, and Paulo Esteves-Verissimo, Fellow, IEEE

Abstract—The advent of next-generation sequencing (NGS) machines made DNA sequencing cheaper, but also put pressure on the genomic life-cycle, which includes aligning millions of short DNA sequences, called reads, to a reference genome. On the performance side, efficient algorithms have been developed, and parallelized on public clouds. On the privacy side, since genomic data are utterly sensitive, several cryptographic mechanisms have been proposed to align reads more securely than the former, but with a lower performance. This manuscript presents *DNA-SeAl* a novel contribution to improving the privacy \times performance product in current genomic workflows. First, building on recent works that argue that genomic data needs to be treated according to a threat-risk analysis, we introduce a multi-level sensitivity classification of genomic variations designed to prevent the amplification of possible privacy attacks. We show that the usage of sensitivity levels reduces future re-identification risks, and that their partitioning helps prevent linkage attacks. Second, after extending this classification to reads, we show how to align and store reads using different security levels. To do so, *DNA-SeAl* extends a recent reads filter to classify unaligned reads into sensitivity levels, and adapts existing alignment algorithms to the reads sensitivity. We show that using *DNA-SeAl* allows high performance gains whilst enforcing high privacy levels in hybrid cloud environments.

I. INTRODUCTION

DNA sequencing and the alignment of sequences are at the heart of applications such as precision medicine, forensics, medical or anthropological research [1], [2], to name a few. Next-generation sequencing (NGS) machines greatly improved the throughput of human DNA sequencing and thereby reduced the costs of DNA analysis to almost 1000\$ per genome. Widely deployed sequencing machines (e.g., from Roche or Illumina) produce short sequences of nucleotides ranging from 30 to 100 base pairs (bp) with error rates around 0.1% [3]. A simplified genomic workflow can be presented as follows. First, the raw sequences of nucleotides that sequencing machines produce, called reads, are aligned to a reference genome to obtain their location in the genome. Then, aligned reads are used as input of the variant calling step, which identifies the donor's genomic

variations (i.e., her/his genotype). Finally, subsequent research or medical applications compare a subject's genotype with other genotypes, or simply study it at a given locus.

On the one hand, data leaks threaten the privacy of human genomes. For example, not only do genomes uniquely identify their owner but they also reveal information about his/her relatives. In addition, once a genome has been revealed, its privacy cannot be recovered, as a subject's genome barely evolves during his life. As multiple studies have shown, anonymizing human genomes, or creating aggregates, cannot fully enforce privacy. To name a few, published privacy attacks included re-identification attacks [4], and disease-revealing attacks [5]. It has also been shown that leaking raw reads can expose their donors to data identity leakage [6]. Consequently, besides standard encryption-based solutions, several works have been published to protect the advanced uses of genomic data: masking information in aligned reads [7], creating privacy-preserving releases of aggregated data [8], classifying raw genomic data as sensitive or non-sensitive [9].

On the other hand, efficient workflows are required, due to the decrease of the sequencing prices that has led to larger data productions and processing workloads. Anticipating this, several works [10]–[12] argued for distributed and high-performance environments to host genomic workflows. Global Alliance [13] developed an ecosystem of worldwide databases that can be remotely accessed. Despite these efforts, patient-derived health data are not widely shared [14]. Recently, several ecosystems addressing the privacy and performance challenges of accessing genomic data in the cloud have been developed. For example, NGS-Logistics [15] allows researchers to analyze rare genomic variants while preserving the privacy of donors. In particular, it relies on different levels of access rights for better protection of the data. However, concerning the processing of genomic data, scientist are considering the use of clouds [16], even though practical privacy-preserving processing of early genomic data has not been defined.

To summarize, privacy attacks on genomic data alerted about the need to incorporate security measures into existing genomic workflows. However, existing cryptographic solutions cannot sustain the high throughput of modern sequencers. Consequently, the status quo is still to rely on plaintext methods, preferentially on private clouds but also on public clouds, such as Amazon AWS¹. For genomic processing workflows to rely on the cheap, widely available and highly efficient public clouds, there is a need for mechanisms that establish a stronger balance between privacy and performance. Indeed, bullet-proof security does not exist, and public clouds may suffer from data leaks caused by internal or external adversaries [17], [18].

This work is partially supported by the University of Luxembourg - SnT and by the Fonds National de la Recherche Luxembourg (FNR) through PEARL grant FNR/P14/8149128, and by the Fundação para a Ciência e para a Tecnologia (FCT) through funding of the LASIGE Research Unit, ref. UID/CEC/00408/2019 and DeST: Deep Semantic Tagger project, ref. PTDC/CCI-BIO/28685/2017.

M. Fernandes, J. Decouchant, M. Völz and P. Esteves-Verissimo are with SNT - Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, 4364 Esch-sur-Alzette, Luxembourg. (e-mail: maria.fernandes@uni.lu, jeremie.decouchant@uni.lu, marcus.voelp@uni.lu, paulo.verissimo@uni.lu).

F. M. Couto is with LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal (e-mail: fjcouto@ciencias.ulisboa.pt).

¹See <https://aws.amazon.com/solutions/case-studies/illumina/>.

In this work, we focus on protecting sensitive genomic data as soon as they are produced by an NGS machine, i.e., before the genomic variations they contain have been determined, and continue to do so throughout the alignment step. As previous works [15] argued, classifying genomic data as either sensitive or not sensitive at all is not sufficient. Some studies support the need for sensitivity degree classifications for genomic and clinical data [19]. Building on this fact, we remarked that a finer grained sensitivity classification of raw reads combined with alignment algorithms that have different privacy guarantees and efficiencies, has the potential to improve the performance and overall privacy of alignment.

As such, *DNA-SeAI* makes the following contributions:

- We present a classification of raw reads into sensitivity levels, based on qualitative and quantitative characteristics of genomic variations. These sensitivity levels are then further partitioned in such a way that an adversary observing a part of the reads of a given sensitivity level, thanks to a successful attack, is not able to infer any more sensitive information from it. We disconnect sensitivity levels, based on the linkage disequilibrium (LD) of genomic variations, and on MaCH [20], a state-of-the-art haplotype inference software.
- Building on previous work, namely [9], we propose to use a detection method based on Bloom Filters (BFs) to efficiently classify raw reads into partitions of sensitivity levels. In particular, we show how to preserve the disconnection property of sensitivity levels when Bloom filters produce false positives among the same or different sensitivity levels.
- We show that given a realistic heterogeneous and distributed environment, one can rely on the diversity of the existing alignment procedures to optimize the privacy \times performance product of the read alignment step.

Whenever a public cloud is available, and is at least as powerful as the private infrastructure, our performance evaluation shows that *DNA-SeAI* requires on average 0.29 CPU seconds and only 1.6 KB of data transfer to securely align a read. Compared to a exclusive public cloud approach, this represents a 10^6 -fold reduction of the computing time and 10^7 -fold reduction of the amount of data transferred to the cloud.

Related Works

The publication of privacy attacks [6], [21] and the use of public cloud environments for biomedical data analysis has raised security concerns. The most widely known privacy attacks perform re-identification of donors, relying on inference techniques and different kinds of personal information [4], [5]. It has also been shown that partial genomic data leaks may enable trail attacks [22], which identify an individual thanks to his unique distinguishing features.

Protecting biomedical data is now a priority challenge for the biomedical community [23]. In order to address this challenge, the biomedical community invested on strategies to protect genomic data privacy and defined three categories of protection: data de-identification [22], data augmentation [24], and cryptographic-based methods [25]. Data de-identification methods remove personal identifiers, such as names and social security numbers, from genomic data, for example through pseudonymization [26]. However, these methods used alone have been shown not to be sufficient to prevent re-identification attacks [22]. Second, data augmentation methods

rely on generalization to achieve privacy protection, basically making records more similar to each other. These methods achieve protection at the price of a lower data utility. Finally, cryptographic-based methods allow users to maintain data utility while protecting the data privacy [25], but they are of limited applicability.

Efficient plaintext alignment methods have been developed, and can be used in parallel in public clouds (either with or without encrypting the data transfers) to study large amounts of data. However, these highly optimized methods, which include CloudBurst [27] and DistMap [28], are not privacy-preserving. Secure alignment algorithms have also been developed, for example using garbled circuits [29] or homomorphic encryption schemes [30], however, they suffer from poor performance. Finally, recently, researchers have been searching for approaches that combine high performance and privacy. Chen et al. [31] proposed a seed-and-extend alignment method, where the seeding step is executed in a public cloud based on keyed-hashes, and the extension step runs in a private cloud. Differently, Balaur [32] makes use of Locality Sensitive Hashing (LSH), secure k-mer voting, and a MinHash algorithm, and Maskal [33] relies on a read filter and Intel SGX enclaves.

Ayday et al. [7] proposed to store encrypted reads in a biobank that enables classified people (e.g., data analyst in a hospital) to retrieve a subset of the reads from a biobank to perform genetic tests while keeping the nature of the tests private. In this approach, the biobank masks parts of the reads, for example, those located outside the request range, or those that the patient did not agree to share. Filtering approaches that identify potential genomic variations at the reads or at the nucleotides level have been described [9], [34]. However, contrary to *DNA-SeAI*, those approaches do not support sensitivity levels. Concurring with our position, of classifying genomic data into sensitivity levels, Dyke et al. [19] proposed a Data Sharing Privacy Test to distinguish degrees of sensitivity for the GA4GH Beacon Project to facilitate data sharing.

Several works [35], [36] determined panels of less than 100 common SNPs, which are sufficient to uniquely identify a subject with very high probability. These panels are often population specific, and made of SNPs carefully selected, for example, based on their minor allele frequency, or linkage disequilibrium (LD). Our method is particularly effective in this situation, where there is an obvious and known sensitivity differential. Considering the inevitable risk residing in use of public clouds, and given the purpose of reducing the risk as much as possible, we introduce a methodology which can be parameterized by: (i) protecting the collections of SNPs that reveal the most information, to a higher standard; (ii) further including the SNPs used in those critical panels in the highest sensitivity levels, to further complicate re-identification.

II. METHODS

A. Data, system and threat model

Data. We build the sensitivity levels based on the genomes of the 1000 Genomes Project [37], and recombine genomic variations with the reference genome GRCh38.p11 to create sensitive sequences that are used to initialize read filters [9],

and classify reads into sensitivity levels. We analyse the sensitivity of reads of 100 and 1,000 bases. These two lengths were selected as representative lengths for short and long reads, accordingly to the existing sequencing machines [38]. We also evaluate the genomic privacy metric on the sensitivity levels using genomes from the 1000 Genomes Project. To study linkage risks, we measure Likelihood Ratio values based on the 2017 iDASH contest dataset², which contains vcf files for 1,000 case and 1,000 control individuals, with 5,511,698 distinct SNPs from chromosome 1.

System model. We consider a biocenter whose task is to generate reads from an individual, and to align those reads to a reference genome in a privacy-preserving manner. To do so, the biocenter can rely on a private cloud, and on several public clouds, which receive an equal random proportion of the reads contained in a sensitivity level. We assume that the sequencing machine and the private cloud are secure. However, we consider a private cloud expensive to maintain, which encourages the use of public clouds even though we assume that the user does not have a complete control over its own data (i.e., which machines are used, etc.) in public clouds. Finally, we consider that all parties rely on encrypted communications.

Threat model. We assume an honest-but-curious adversary, who tries to observe sensitive genomic information during the alignment of reads. In particular, the adversary is able to observe raw reads in the public cloud if they are used in plaintext alignment algorithms. However, we assume that no more than $f = 1$ public clouds can be compromised, so that privacy guarantees increase with N/f , where N is the total number of clouds used. We also assume that the adversary has access to a reference genome, is able to align raw reads to obtain the biological insights they contain, and has access to the statistical relationships between genomic variations. Obtaining such refined data can then potentially enable existing privacy attacks during future uses of data (e.g., if allele frequencies in a case population are publicly released). To limit the risk that an adversary obtains sensitive information, while obtaining high performance using cleartext alignment, we use a risk-threat approach to protect the reads.

Privacy goals and amplification attack. In addition, as perfect security does not exist, in case a successful attack happens, where an adversary would be able to observe raw reads, *DNA-SeAI* aims at preventing this attack from being extended to data that could not be observed directly during the attack (e.g., because it is more protected, or used in a different location). We call this an amplification attack. First, the possible presence of a rare allele for a given locus cannot be inferred from the observation of a single common allele in a compromised cloud, since at most one public cloud is assumed to be compromised, and could therefore host a second common allele. Then, we use Linkage Disequilibrium (LD) measurements, and MaCH [20], a state-of-the-art haplotype inference software, to make sure that an adversary is never able to execute such amplification attacks. Overall, our approach complicates future re-identification attacks, since significant information is harder to obtain, as measured through the genomic privacy metric, and we show that linkage attacks are prevented when levels are splitted across several clouds.

B. Sensitivity levels

We now describe how to create the different sensitivity levels to prevent privacy leak amplifications, namely using promotions of genomic variations across levels.

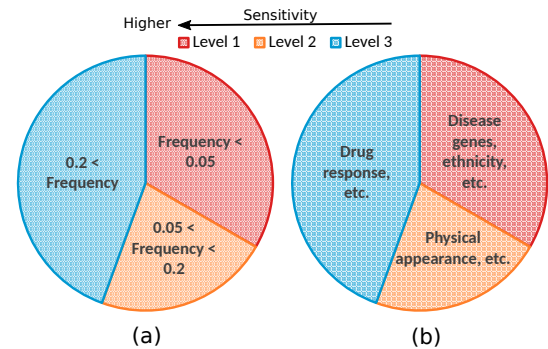


Fig. 1. Initial sensitivity levels based on: (a) alleles frequency (quantitative classification); and (b) manual declarations (qualitative classification).

a) Qualitative and quantitative sensitivity levels: In this manuscript we propose the creation of sensitivity levels that allow the differentiation of genomic data and basically can be a combination of two different methods: manual declaration (qualitative), or based on frequencies in a reference population (quantitative). In the first case, a system administrator defines the sensitivity levels based on how he perceives the sensitivity of the information a variation reveals. Sensitive levels based on frequencies, as we propose in this manuscript, are built on the fact that a rare disease/genetic variation should be considered more sensitive than a common one, since they concern a smaller subset of the population. In particular, alleles whose frequencies are lower than 0.05 should always be considered highly sensitive, since they can lead to a restricted group of individuals [39].

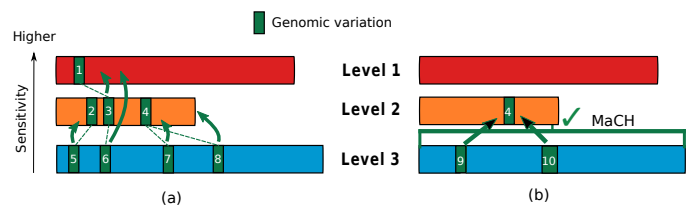


Fig. 2. Alleles promotion across sensitivity levels based on linkage disequilibrium (LD).

Figure 1a shows, as an example, a distribution of genomic variations in three sensitivity levels, based on the frequency of the alleles in the genome. In this figure, level 1 contains the alleles whose frequency goes up to 0.05, level 2 is composed of the alleles whose frequency is comprised between 0.05 and 0.2, and level 3 holds the remaining more common alleles. Later in the results section, we discuss the distribution of the alleles among different possible sensitivity levels. Differently, Figure 1b presents three sensitivity levels based on information coded in the genome and their severity when leaked. For level 1 we consider information such as disease genes, ethnicity-related variants and other regions that can lead to individual's re-identification. In this article, we focus on quantitative levels, and leave the implementation of qualitative levels for future work.

²<http://www.humangenomeprivacy.org/2017/>

b) Sensitivity level promotion through LD connections: Linkage disequilibrium (LD) describes the non-random transmission of genomic variations. These non-random associations have been used in privacy attacks [6]. We compute the LD between two genomic variations if they are located at a maximum distance of 1000 bases. Figure 2a illustrates how we prevent adversaries from using LDs to infer genomic variations through sensitivity levels. As an example, we show a variation 1 linked with variation 3, which is in turn linked to variation 6. In this case, we have direct inference connections between all the three sensitivity levels. For example, an adversary obtaining the variation 3 would be able to infer the variations 1 and 6. To avoid such an attack, we promote variations 3 and 6 to level 1 (represented by the green arrows), which prevent the attacker from obtaining all the variations (1, 3, and 6). Following this procedure, we also promote variations 5 (linked to variation 2) and variations 7 and 8 (linked to variation 4) to level 2.

c) Sensitivity level promotion through haplotypes inference: To ensure that sensitivity levels are completely disconnected, we also use a haplotype inference software (e.g., MaCH [20]). Figure 2b shows how we processed in the case of inference relations between variations. We run MaCH on the SNPs from one sensitivity level to determine which SNPs can be inferred with an r^2 value higher than 0.3 (as recommended by MaCH's authors) from higher sensitivity levels. The example in the figure shows that an adversary obtaining the variations 9 and 10 would be able to infer variation 4. Therefore, to prevent it we promote variations 9 and 10 to the same level of variation 4. The software receives as input two sets: (i) a list of the biomarkers and the SNPs information of a reference population; and (ii) a list of the biomarkers and the SNPs information the adversary can observe.

We perform the following steps for each sensitivity level:

Step 1. We start by creating sets of four files based on all subsets of 20k SNPs from Chr. 1, available in the 1000 Genomes Project. We believe this number to be large enough, since the correlation of SNPs decreases with their distance. The first two files details the set of genotypes the adversary uses as reference. The second set of files details the genomic variations the adversary can observe at a given sensitivity level (through an hypothetical attack) and does not contain the more sensitive SNPs, which would be located in a secure environment.

Step 2. We run MaCH with the input files, and obtain a list of SNPs that can be inferred by an adversary with the provided sets, i.e., masked SNPs that MaCH is able to discover with good accuracy. We focus on the inference of more sensitive SNPs, thus we ignore those that are inferred and belong to the same sensitivity level. This step assesses the information that can be inferred in case of information leakage from a sensitivity level.

Step 3. From each of the inferred SNPs, we compute the top 10 most connected SNPs (through LD) that the adversary can observe. We therefore remove at most 10 SNPs per inferred SNP at each iteration.

Step 4. We remove the 10 most related SNPs from the initial adversary set, and move then to the higher sensitivity level, since they allow the inference of more sensitive SNPs. We then reiterate the whole process, starting from Step 1 using the newly obtained input files, until no more inferences are possible, or their number stabilizes.

Figure 2b provides an illustration of the inference iteration process using MaCH after alleles in strong direct LD have been promoted. In this example, from the genomic variations contained in sensitivity level 3 (created during step 1), MaCH tries to infer more sensitive genomic variations (step 2). After inference, MaCH infers that the genomic variation numbered 4, which is in level 2, can be inferred with good accuracy from those in level 3. Our code would then identify that the genomic variations numbered 7 and 8 in level 3 are strongly associated with the one numbered 4 in level 2 (Step 3). Those two genomic variations would then be promoted to level 2 (Step 4), before iterating the inference process.

C. Classifying sensitive reads and adapted treatment

In this section, we first recall how sensitive reads (i.e., reads that carry sensitive personally identifiable information (PII)) can be detected thanks to a filtering method first presented by Cogo et al. [9]. We then extend this method to classify reads into several sensitivity levels depending on the magnitude of the impending privacy risk – the probability of an adversary extracting PII, and the resulting negative impact. Finally, we explain how to solve possible detection conflicts when using several filters.

Reads filter. We briefly introduce the reads filtering method [9], [34] and how we can use it to classify reads into a scale of sensitivity levels. The reads filters are implemented using Bloom filters, which are high throughput data structures that can produce false positives, but never false negatives. In addition, the filters are not a bottleneck when used in combination with sequencing machines, as they are always at least 40 times faster than current NGS machines, and they are parallelizable. The filters are initialized from a database of reads known to carry sensitive information. This sensitive information includes, but is not limited to, all existing data that have been used in the literature that describe attacks to re-identify subjects of experiments. Such attacks have been based on three kinds of sensitive sequences: (i) genomic variations (including SNPs and SVs), (ii) disease genes and (iii) short tandem repeats (STRs) – a known small string that appears several times contiguously in a subject's DNA, and whose repetition numbers vary among a population.

Classification into sensitivity levels. We use one short read filter per sensitivity level that we have previously identified, to prevent amplification attacks. To build the dictionary of sensitive levels, we collect all genomic variations in the corresponding disconnected sensitivity levels, and create all possible short genomic sequences that contain them, relying on the genomic variations database and the reference genome. Each of those sequences is then finally inserted into a Bloom filter, which is then ready to filter sequences. Figure 3 illustrates our filtering approach, which classifies reads according to their sensitivity level. The filtering approach is done in two steps: sensitivity-aware filtering, and conflict management, which is required when a read matches in several filters. After the sequencing step, the reads are given as input to the sensitivity-aware filtering step, made of Bloom filters initialized with several disconnected groups of genomic variations. Finally, the conflict management step combines the output of the filters to determine the sensitivity of reads. It is then possible to adapt the storage, computation, and use of a read according to the

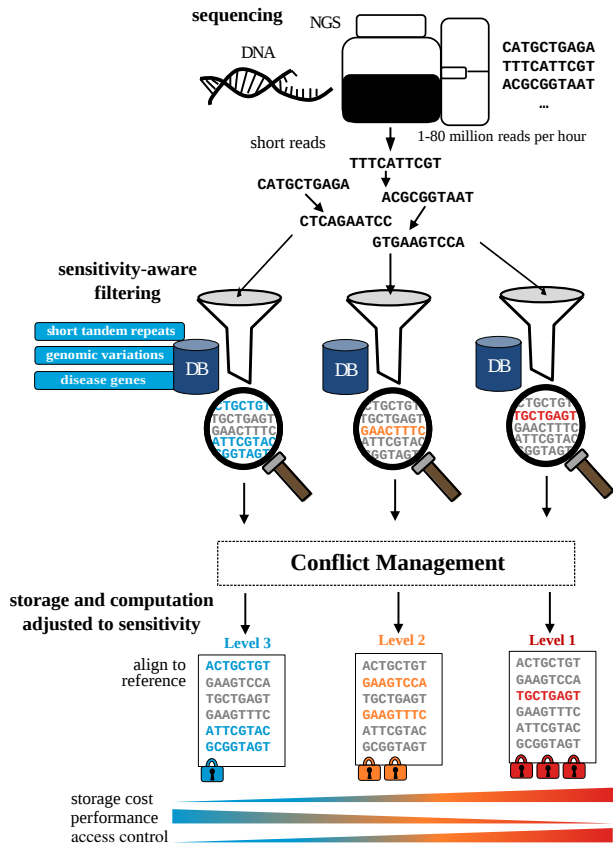


Fig. 3. Classification of reads in sensitivity levels and adjusted storage, algorithms, and access control per sensitivity level.

security it requires. As represented in Figure 3, storage costs and access limitations per read tend to increase with the sensitivity of reads, while alignment performance tends to decrease. Specific numbers depend on the available infrastructures and on design choices. To prevent the linkage of a set of reads to a disease, we randomly distribute the reads of each sensitivity level among several clouds.

Filters conflict management. Bloom filters may produce a controlled number of false positives, which in our case may cause privacy leaks if they are not correctly managed. Handling false positives is part of the conflict management step represented on Figure 3, and works as follows. If a read matches in several sensitivity level filters, we set its sensitivity to the level of the most sensitive filter it matches to.

D. Read alignment: Performance \times Privacy optimization

We finally show that the performance \times privacy product of reads alignment is improved when adapting the alignment algorithms used, along with their execution environment, depending on the detected sensitivity of the reads as soon as the public cloud is at least as powerful as the private cloud. More specifically, to align reads depending on their detected sensitivity levels, *DNA-SeAl* aligns the most sensitive reads in the private cloud (using CloudBurst [27]), and the least sensitive reads in the public cloud (using CloudBurst [27]). Depending on the scenario studied, possibly remaining reads can be aligned either in the private cloud (using CloudBurst [27])

or in the public cloud (using Chen et al.'s protocol [31]), depending on which cloud finishes first.

We study the resulting performance improvement of our approach over standard alignment strategies:

- *Private clouds only.* A biocenter relies entirely on its private secure infrastructure to align reads using unsecure algorithm.
- *Public clouds only.* Alignment is performed in a non-secure environment where an adversary may observe unencrypted computations and communications. Therefore, sensitive reads are aligned with proven or believed secure algorithms, while more efficient algorithms are used with low sensitivity reads.
- *Sensitivity-adapted private and public clouds alignment.* High-sensitivity and low-sensitivity reads are aligned in private and public clouds, respectively. This scenario makes a rational usage of a biocenter's computing resources for the sensitive computations, extended by a secure usage of public clouds.

III. RESULTS

A. Sensitivity levels statistics

We studied the average proportion of a subject's SNPs in each sensitivity level before and after SNPs promotion through haplotype inference. Figure 4a represents the proportion of genomic variations of a subject in each sensitivity level before the promotions. Level 1 contains a minority of alleles (3%), level 2 contains only 2% of the alleles, and the remaining 95% lies in level 3. The genomic variations promotion slightly change the distribution among the sensitivity levels, as Figure 4b shows. In this case level 1 is the smallest one with 3%, level 2 slightly increases with now 7% of the alleles, and the last, level 3, contains 90% of the alleles.

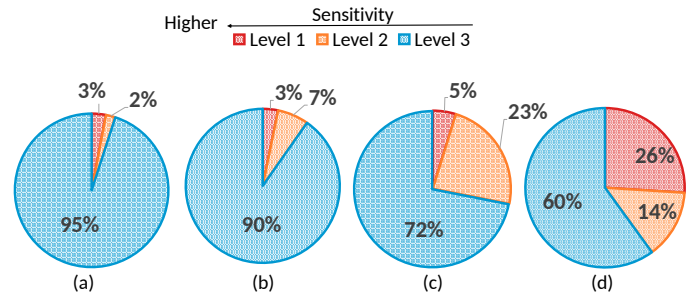


Fig. 4. Evolution of the sensitivity levels through the alleles promotions: (a) Initial proportion of an individual genomic variations (GVs) per sensitivity level; (b) Proportion of an individual GV's per sensitivity level after promotions; (c) Proportion of 100-bases reads per sensitivity level; (d) Proportion of 1000-bases reads per sensitivity level.

B. SNPs promotion across sensitivity levels

After one iteration, we promoted 1.6% of the SNPs of level 3 to level 2, and 18% of the SNPs of level 2 to level 1. Overall, we promoted 1.5% of all SNPs from one level to a more sensitive one. We summarize the proportion of inferred SNPs per sensitivity level after various rounds of promotion iterations, in Table I. The promotions are made using the method described in section II-B. After one inference iteration with MaCH, very few genomic variations could still be inferred (e.g., less than 5 SNPs with level 3). These inferences are due to the limited number of genomes used in the 1000 Genomes project, and to specific individuals who had unique

combinations of statistically unlinked SNPs (since they can still be inferred after more iterations). We are confident that inferring those few SNPs in a larger population would not be possible because the number of unique combinations of several SNPs would be rarer.

Promotion iterations	0	1	2	3	4	5
Inferred SNPs (%) from level 2 to level 1	9.97	0.05	0.05	0.05	0.05	0.05
Inferred SNPs (%) from level 3 to level 2	0.43	0.30	0.14	0.10	0.06	0.04

TABLE I. NUMBER OF INFERRER SNPs PER INFERENCE AND PROMOTION.

C. Reads classification in sensitivity levels

Figure 4c shows the classification of 100-bases reads in each sensitivity level. This distribution is somewhat different from the distribution of the genomic variations. Level 1, which is the most sensitive level, only contains 5% of the reads, while level 2 contains 23% of the reads, and the remaining 72% of the reads are classified into level 3. Level 3 continues to hold the majority of the information which support the performance and privacy optimization we discuss in next section. Figure 4d shows that 60% of the 1000-bases reads are classified into level 3, while 14% are in level 2, and the remaining 26% are in level 1. The overall sensitivity of long reads is higher, since, on average, they contain more SNPs.

D. Privacy evaluation

We evaluated the increased privacy protection that the use of sensitivity levels can bring to genomic data using two metrics: the genomic privacy metric, and the Likelihood Ratio (LR) value. Genomic privacy represents the weighted risk of re-identification based on adversary estimates for the minor allele of observed SNPs. For this metric, low values indicate high privacy [40]. The LR value represents the upper bound power for the detection of an individual in a case group [41].

Fig. 5 shows the genomic privacy values for individual HG03556 computed on its alleles for each sensitivity level and per chromosome. The most sensitive level (red line) has high genomic privacy values for all chromosomes (mainly between 7.5×10^4 and 7.5×10^5), and it represents the largest contribution to an individual's genomic privacy value. The values for the intermediary level (orange line) are comprised between 3.0×10^4 and 2.0×10^5 , and the least sensitive level (blue line) is evaluated at less than 10^5 . This experiment shows that the SNPs increasingly participate in the genomic privacy metric as they are classified more sensitive. We observed the same pattern on the complete 1000 Genomes Project population.

We then evaluated the LR-test metric on the case and control individuals considering 2,014,777 SNP sites. Overall, depending on the subject, level 1 contained 130,000–158,000 alleles, level 2 contained 111,000–125,000 alleles, and level 3 contained 1,733,000–1,773,000 alleles. We randomly partitioned each sensitivity level such that no individual could be identified as being part of the case population from any partition's subset. Our experiments indicate that splitting levels 1 and 2 in half (i.e., 50% subsets), and level 3 in five (i.e., 20% subsets) would prevent any successful linkage. We ran

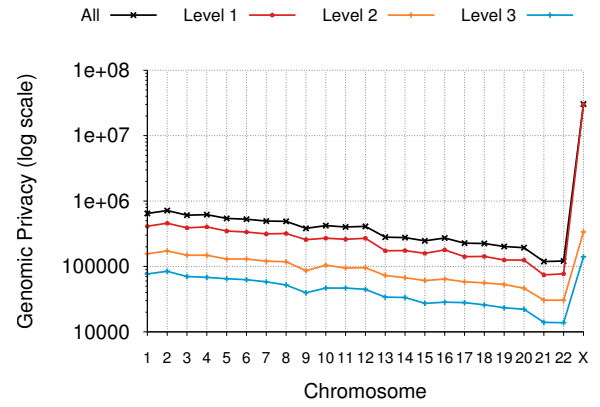


Fig. 5. Genomic privacy per sensitivity level for individual HG03556.

each experiment 20 times with all 1,000 case and 1,000 control genomes.

Fig. 6 shows the distribution of the LR-test values for the full sensitivity levels and for random subsets of the sensitivity levels. In this figure, we denote the case population as ca and the control population as co. The numbers in the figure represent the proportion of case individuals that can be identified without false positives. Using the full genomic information, 31.3% of the case individuals could be linked with the disease. Based on the full sensitivity levels, 32.9% (i.e., 329 individuals), 6.2% and 3.7% of the case individuals could be linked with the disease based on the information contained in levels 1, 2 and 3, respectively. We then partitioned the per-subject levels until no case individual could be linked with the disease based on any partition subset. Dividing levels 1 and 2 in half (50%) was sufficient to achieve this goal, while level 3 required to be partitioned in five (20%). This result experimentally shows that randomly partitioning levels of reads allows the processing of sets of variants while preventing linkage attacks.

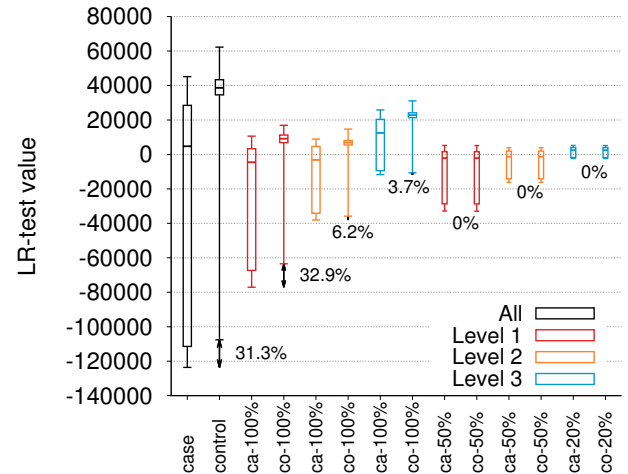


Fig. 6. Distribution of the LR-test values per sensitivity level subsets.

E. Performance × Privacy product optimization

Aligning reads in a cloud implies assuming that the cloud provider is trustful, and that the cloud will not be attacked. We do not make these risky assumptions, and therefore rely on the

following three categories of alignment algorithms, which we previously introduced in Section I, to optimize the performance \times privacy product using sensitivity levels. Category (i) – *non-secure but fast algorithms*: we use Cloudburst [27], which is an unprotected method, and requires 0.4 CPU seconds, respectively 0.41 CPU seconds if reads are encrypted for the transfer to the cloud server. Category (ii) – *secure but slow cryptographic algorithms*: we use a homomorphic encryption based approach [42], which requires 22 CPU days. Category (iii) – *algorithms providing an intermediate level of protection*: we use a protocol based on hashing k-mers presented in [31], which is much more efficient, requiring only 1.3 CPU seconds. However, it may leak information about equal k-mers and has not been formally proven secure. Table II summarizes our analysis of the privacy level, computation cost (CPU hours) and communication (bytes) cost of aligning a single 100 base-pairs read to the full human genome, using a single core.

Method	Privacy (Sec. II-C)	Computation (CPU time)	Communication volume
Homom. encr. [42]	Very high	22.08 days	3.75×10^8 KB
Hashed k-mers [31]	High	1.3 sec.	5.22 KB
Cloudburst [27]	Low	0.41 sec.	2.3 KB

TABLE II. PRIVACY, PERFORMANCE AND COMMUNICATION OVERHEADS OF THE ALIGNMENT ALGORITHMS WE USE.

Table III lists situations with different relative proportions of public cloud's computing power over the private cloud computing's power. For example, configuration 1/10 means that the public clouds are 10 times more powerful than the private cloud. Under each configuration we evaluate the performance of a read alignment for the three possible cases: (i) on the public cloud only (using 5PM [42]); (ii) on the private cloud only (using CloudBurst [27]); or (iii) on both the private cloud for sensitive reads (using CloudBurst [27]) and on the public clouds for non-sensitive reads (using 5PM [42]).

Proportion Pub./Pri.	Our approach	Previous approaches		
		Pub [42]	Priv [27]	Pub. [42]/Pri. [27] with [9]
	Time	$(3 \times 10^8$ s)	(0.41s)	(0.29s)
1/1	0.29s	10^6 x	1.39x	1x (0.29s)
2/1	0.097s	10^6 x	4.20x	1.51x (0.11s)
10/1	0.019s	10^6 x	20x	5.85x (0.11s)
	Data Transfers	(16.8GB)	(2.3KB)	(1.6KB)
1/1	1.6KB	10^7 x	1.39x	1x (1.6KB)
2/1	0.55KB	10^7 x	4.20x	1.51x (0.83KB)
10/1	0.11KB	10^7 x	20x	5.85x (0.65KB)

TABLE III. OVERHEADS OF EXISTING PRIVACY-PRESERVING ALIGNMENT APPROACHES COMPARED TO *DNA-SeAI*'s, DEPENDING ON THE RATIO OF PUBLIC/PRIVATE CLOUDS.

Overall, we can draw the following conclusions about privacy-preserving alignment of a read: (i) it is not practical to rely only on a public cloud to align reads with cryptographically secure algorithms (3×10^8 seconds per CPU per read); (ii) relying on a private cloud with cleartext alignment is the fastest solution (0.41 seconds per CPU), but it does not scale; (iii) by classifying reads as either sensitive or non-sensitive, performance can be improved whenever a public cloud, assumed to be as least as powerful as the private cloud, is available (starting at 0.29s with [9]); and (iv) our approach, relying on more than two sensitivity levels, further improves

performance on hybrid clouds (down to 0.019s with a public cloud ten times more powerful than the private cloud). Similar conclusions can be taken in terms of memory consumption. To summarize our approach, in a nutshell, and compared to previous works, by *using sensitivity levels to align reads, we remove computational tasks from the secure alignment performance bottleneck (i.e., the private cloud alignment), and execute them securely in public clouds.*

IV. DISCUSSION

In this manuscript we presented *DNA-SeAI*, which makes the following contributions:

- We proposed a methodology to create sensitivity levels for unaligned reads. Our methodology allows sensitive levels to be defined based on both qualitative and quantitative aspects. The levels declared qualitatively are based on the biological insights a sequence reveals, while the levels declared quantitatively are based on the frequency of carried genomic variations in a population. We based our experiments on quantitative levels only, for simplicity, since qualitative levels are both subjective and of relatively small size.
- We found out that leakage across levels exist due to haplotype inference (using LD relations), and showed that promoting groups of linked genomic variations to the highest of their sensitivity levels prevent such leakages.
- We extended the short reads filter proposed by Cogo et al. [9] to automatically classify reads into the multiple sensitivity levels (i.e., not just based on a binary answer).
- We defined a read alignment method that relies on a classification of reads into sensitivity levels, which improves over the state-of-the-art alignment methods in terms of performance while providing adequate security guarantees.

Filtering limitations. The filter cannot detect genomic variations it was not initialized with (e.g. de novo SNPs).

However, new genomic variations are now more rarely discovered [9], which limits the residual risk of not detecting sensitive nucleotides. Moreover, the filters can be very easily updated to include newly discovered genomic variations. In a production system, this would be as straight forward as anti-virus update schemes today.

Parameters. We define three sensitivity levels, however, this number can be extended as more diverse algorithms and execution environments are available. Relying on more levels can increase: i) performance while maintaining a given security; or ii) security while maintaining a given performance. We made practical choices concerning the parameters of our methods. During the SNP promotions based on LD, we used 20,000 as a maximal distance between SNPs during LD computations, which provides a good accuracy, since most of the reported haplotype blocks in humans are smaller than 20Kbases [43]. During the inference step, which relies on MaCH, we promoted, during each inference iteration, the 10 SNPs the most connected to an inferred SNP. Promoting less SNPs per iteration would result in less promotions overall but would take a larger number of iterations. In future work, we will consider refining the sensitivity levels based on ethnicities.

V. CONCLUSION

In this manuscript, we proposed a novel approach to classify genomic data in multiple incremental sensitivity lev-

els. We explained how to disconnect these levels based on LD relations, and how to prevent attacks amplification. We showed that such classification leverages the complementary characteristics of different alignment algorithms, if selectively applied to subsets of the data reads, guided by such a risk-aware sensitivity classification, taking the best of each algorithm (performance or security). Our approach, *DNA-SeAI*, improves on the state of the art in terms of privacy \times performance product, taking into account the computation time and communication cost to the clouds. Furthermore, *DNA-SeAI* is suitable for different levels and different algorithms, even as new algorithms appear. We presented an implementation with multiple filters that efficiently and automatically classify unaligned reads in privacy-sensitivity levels. This filtering approach allows adjusting the protection of reads of different levels, with incremental performance gains resulting in an optimized and stable privacy \times performance product. We show that the filtering approach can be combined with existing alignment methods (either cleartext, hybrid, cryptographic). We believe *DNA-SeAI* to be timely, presenting a necessary tradeoff between perfect security and performance, since the growth in genomics data production pushes biocenters to rely on public clouds, and since the performance of cryptographic approaches is not sufficient to be massively used. Finally, *DNA-SeAI*'s classification reduces the future re-identification risks and the partition of the levels prevents linkage attacks.

REFERENCES

- [1] M. West, G. S. Ginsburg *et al.*, "Embracing the complexity of genomic data for personalized medicine," *Genome Res.*, vol. 16, no. 5, pp. 559–566, 2006.
- [2] M. Kayser and P. de Knijff, "Improving human forensics through advances in genetics, genomics and molecular biology," *Nat. Rev. Genet.*, vol. 12, no. 3, pp. 179–192, 2011.
- [3] M. L. Metzker, "Sequencing technologies the next generation," *Nat. Rev. Genet.*, vol. 11, no. 1, pp. 31–46, 2010.
- [4] M. Gymrek, A. L. McGuire *et al.*, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.
- [5] D. R. Nyholt, C.-E. Yu *et al.*, "On jim watsons apoe status: genetic information is hard to hide," *Eur. J. Hum. Genet.*, vol. 17, pp. 147–149, 2009.
- [6] R. Wang, Y. F. Li *et al.*, "Learning your identity and disease from research papers: information leaks in genome wide association study," in *CCS*, 2009, pp. 534–544.
- [7] E. Ayday, J. L. Raisaro *et al.*, *Privacy-preserving processing of raw genomic data*, 2014.
- [8] X. Zhou, B. Peng *et al.*, "To release or not to release: evaluating information leaks in aggregate human-genome data," in *ESORICS*, 2011.
- [9] V. V. Cogo, A. Bessani *et al.*, "A high-throughput method to detect privacy-sensitive human genomic data," in *WPES*, 2015.
- [10] P. E. Verissimo and A. Bessani, "E-biobanking: What have you done to my cell samples?" *IEEE Secur Priv*, vol. 11, no. 6, pp. 62–65, 2013.
- [11] A. Bessani, J. Brandt *et al.*, "Biobankcloud: a platform for the secure storage, sharing, and processing of large biomedical data sets," 2015.
- [12] B. Liu, R. K. Madduri *et al.*, "Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses," *Journal of biomedical informatics*, vol. 49, pp. 119–133, 2014.
- [13] T. G. A. for Genomics and Health, "A federated ecosystem for sharing genomic, clinical data," *Science*, vol. 352, pp. 1278–1280, 2016.
- [14] P. R. Payne, N. H. Shah *et al.*, *Democratizing Health Data for Translational Research*, 2017, pp. 240–246.
- [15] A. Ardehshirdavani, E. Souche *et al.*, "Ngs-logistics: federated analysis of ngs sequence variants across multiple locations," *Genome Med.*, vol. 6, no. 9, p. 71, 2014.
- [16] B. Langmead and A. Nellore, "Cloud computing for genomic data analysis and collaboration," *Nat. Rev. Genet.*, vol. 19, pp. 208–219, 2018.
- [17] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data in the cloud," in *DSN-W*, 2011.
- [18] A. Duncan, S. Creese *et al.*, "An overview of insider attacks in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 12, pp. 2964–2981, 2015.
- [19] S. O. Dyke, E. S. Dove *et al.*, "Sharing health-related data: a privacy test?" *NPJ genomic medicine*, vol. 1, pp. 1–6, 2016.
- [20] Y. Li, C. J. Willer *et al.*, "Mach: using sequence and genotype data to estimate haplotypes and unobserved genotypes," *Genet. Epidemiol.*, vol. 34, no. 8, pp. 816–834, 2010.
- [21] K. B. Jacobs, M. Yeager *et al.*, "A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies," *Nat. Genet.*, vol. 41, no. 11, pp. 1253–1257, 2009.
- [22] B. Malin and L. Sweeney, "How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems," *J Biomed Inform*, vol. 37, no. 3, pp. 179–192, 2004.
- [23] M. Naveed, E. Ayday *et al.*, "Privacy in the genomic era," *CSUR*, vol. 48, no. 1, p. 6, 2015.
- [24] Z. Lin, M. Hewett *et al.*, "Using binning to maintain confidentiality of medical data," in *AMIA Symposium*, 2002, p. 454.
- [25] M. Kantarcioglu, W. Jiang *et al.*, "A cryptographic approach to securely share and query genomic sequences," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 5, pp. 606–617, 2008.
- [26] T. Neubauer and J. Heurix, "A methodology for the pseudonymization of medical data," *Int. J. Med. Inform.*, vol. 80, no. 3, pp. 190–204, 2011.
- [27] M. C. Schatz, "Cloudburst: highly sensitive read mapping with mapreduce," *Bioinformatics*, vol. 25, no. 11, pp. 1363–1369, 2009.
- [28] R. V. Pandey and C. Schlötterer, "Distmap: a toolkit for distributed short read mapping on a hadoop cluster," *PLoS One*, vol. 8, no. 8, pp. 1363–1369, 2013.
- [29] Y. Huang, D. Evans *et al.*, "Faster secure two-party computation using garbled circuits," in *USENIX*, vol. 201, no. 1, 2011.
- [30] E. De Cristofaro, S. Faber *et al.*, "Secure genomic testing with size-and position-hiding private substring matching," in *WPES*, 2013.
- [31] Y. Chen, B. Peng *et al.*, "Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds," in *NDSS*, 2012.
- [32] V. Popic and S. Batzoglu, "A hybrid cloud read aligner based on minhash and kmer voting that preserves privacy," *Nat. Commun.*, vol. 8, 2017.
- [33] C. Lambert, M. Fernandes *et al.*, "Maskal: Privacy preserving masked reads alignment using intel sgx," in *SRDS*, 2018.
- [34] J. Decouchant, M. Fernandes *et al.*, "Accurate filtering of privacy-sensitive information in raw genomic data," *Journal of Biomedical Informatics*, vol. 82, pp. 1–12, 2018.
- [35] S. Yousefi, T. Abbassi-Dalooi *et al.*, "A snp panel for identification of dna and rna specimens," *BMC Genomics*, vol. 19, no. 1, p. 90, 2018.
- [36] Y.-L. Wei, C.-J. Qin *et al.*, "Validation of 58 autosomal individual identification snps in three chinese populations," *Croatian Medical Journal*, vol. 55, no. 1, pp. 10–3, 2014.
- [37] "1000 Genomes Project: A Deep Catalog of Human Genetic Variation," available at: <http://www.1000genomes.org/>.
- [38] M. O. Pollard, D. Gurdasani *et al.*, "Long reads: their purpose and place," *Hum. Mol. Genet.*, vol. 27, no. R2, pp. R234–R241, 2018.
- [39] S. Sankararaman, G. Obozinski *et al.*, "Genomic privacy and limits of individual detection in a pool," *Nat. Genet.*, vol. 41, pp. 965–967, 2009.
- [40] I. Wagner, "Genomic privacy metrics: A systematic comparison," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 50–59.
- [41] X. Jiang, Y. Zhao *et al.*, "A community assessment of privacy preserving techniques for human genomes," *BMC Medical Informatics and Decision Making*, vol. 14, no. 1, p. S1, 2014.
- [42] J. Baron, K. El Defrawy *et al.*, "5pm: Secure pattern matching," in *SCN*, 2012, pp. 222–240.
- [43] J. D. Wall and J. K. Pritchard, "Haplotype blocks and linkage disequilibrium in the human genome," *Nat. Rev. Genet.*, vol. 4, p. 587, 2003.